LSTM Nvidia Stock Prediction Model

Summer Olmstead
University of Tennessee at Chattanooga
Computer Science and Engineering Department
Introduction to Machine Learning
4430
Chattanooga, TN, U.S.A
yjr285@mocs.utc.edu

***Abstract*— With many spectating the company Nvidia to become one of the most rapid growing companies in the future as it has a key role in manufacturing the superior software chips that many of the upcoming groundbreaking AI infrastructure has already been built around, this study aims to use an LSTM machine learning algorithm to predict Nvidia's stock price 90 days into the future. After creating our model, i can successfully create a prediction which displays a predicted increase in price in the next three months.**

Keywords—Machine Learning, LSTM, Nvidia, Stock Prediction Analysis

I. INTRODUCTION

This paper discusses the use of a Long Short-Term Memory (LSTM) machine learning algorithm for stock prediction on Nvidia's stock. Nvidia is a company mainly known for its leading global manufacturing of high-end graphics processing units(GPUs). Nvidia is one of the more popular stocks, many are watching at the moment because of its predicted success in the nearby future due to their advancements in tech. In this study, I will see from the data the likelihood of this predicted success many are predicting from this company. From gathering the past 5 years of Nvidia's stock history with the features of Open, High, Low, Close prices from Yahoo Finance, I trained and tested the model with this data to predict the stock price of Nvidia forecasting into the next 90 days.

II. MOTIVATION

The motivation for this study is to be able to successfully write a machine learning algorithm which can recognize trends and patterns by looking at the historical data of the stock and use this data to make predictions. For this reason, this is why a Long Short-Term Memory (LSTM) Machine Learning Algorithm was chosen to be the model architecture because it can recognize patterns based on the time sequences it has been trained on. Moreover, with the fast-growing stock history of Nvidia, I am curious if I can successfully implement a forecasting method to see the company's growth over time. A stock price is not only for investors to gain profit, but it also can recognize the well-being of the company over time. This model if successful with accurate predictions, which would only be applicable to judge in the future when the predictions become the past or present, could be an extremely important tool for investors or people who are curious about the forecast of the company. In addition, another factor contributing to my motivation towards the study is if the prediction of this stock is successful, I could begin doing predictions of multiple stocks at once to have a bigger picture of the market and begin including more variables into the predictions to make it more accurate with the market. Some potential future considerations of what could affect the stock market given the economic implications from theory are increase or decrease of government spending, interest rates, increase or decrease of the money supply in the United States, and more. In many economic models, investment is shown to have an impact on any fiscal or monetary policy the government implements. Therefore, if this model is successful the bigger motivation would be to create a bigger network of stock predictions of the market to make the most accurate predictions by even factoring into the model the current economic state and more which affects the stock market. Another ideal consideration into the model would be somehow adding a news factor which would affect the stock price which could be if the company entered a scandal which led people to sell and so on.

III. METHODOLOGY

A. Gathering the Data

First, I gathered the data from Yahoo Finance [1]. Yahoo Finance reports statistics for every day the stock market is open. The stock market is open 252 trading days in a year. On the day I began this project which was April 15, 2024 I downloaded a five year csv file which contained the history of the Nvidia Stock. This file contained everyday since April 15, 2019 until April 15, 2024 with the information of Open Price, High Price, Low Price, Close Price, Adjusted Close Price, and Volume of the Nvidia stock. The open price is the price the stock opened with in a particular day. The high and low prices show the highest and lowest price of the stock on that day. The close price is the closing price of the stock when the market closed on that day. The adjusted close price is the price of the stock after dividends or pay outs. The volume of the stock is how many people are currently trading or invested in that stock on that day. I took this five year history csv fie and loaded it into my code by using the Pandas package to read and put this data into a data frame.

B. Cleaning the Stock Data

As the stock has its numbers reported every day, in the data frame there was no NaN values. However, to add depth of analysis to the model I created a new column into the data frame which was a moving average with a window of 10 days. This is taking the past 10 days of the stock's open price and averaging it to calculate this value. This is a common method for data analytics in stock prediction to capture trends and patterns in the data. I decided to use a lower moving average number of 10 because I thought it would be better to have a general sense of multiple short term predictions when we do eventually make our predictions of 90 days. This would help the model capture short time trends better from data point to data point in predictions. From trial and error, I added this feature to have this ability to capture trends better in the model in general because the model was lacking statistical depth. However, this lead to the issue of the first 10 initial values starting from April 15, 2024 to become a NaN values as they had to previous 10 values to take average of. So I filled those values with the backward fill method which was essentially filling all of them with the 11th day first moving average value. I wanted to ensure that in the data frame there was no NaN values to have to potential to mess up the model. Because now there are no NaN values, I could move onto the next step of creating the LSTM model.

C. Creating the LSTM Model

To create the LSTM Model I first declared the window size to be 90. This is because I wanted the model to be able to capture and recognize trends in the time sequence of 90 days. Moreover, our goal is to predict 90 days into the future so it only made sense to make the window size to be 90. Then from the data frame with the features of Open Price, Volume, and Moving Average 10, I scaled all of these with the goal to feed them normalized into the model and added the scaled versions in their own column in the data frame. Scaling, also known as normalizing, is extremely important to make sure all data contributes to the model equally and this avoids the domination of features with larger values or different units. After this, I created a split of 60% training, 20% testing, and 20% validation. I loaded the data then into its train, test, and validation datasets. I never had made a validation split before this project; however, I found this benefitted the accuracy of the model extremely to include this step.

Next, I created my LSTM model with the use of common PyTorch architecture. With the input parameters of the model being input size, hidden size, number of layers, and output size. I used a dropout rate of 0.2 in my forward method of my init() class. The hyperparameters I chose were to have my input size at 3 as I have 3 features I input into the model. I chose 64 hidden layers. The output size is 3 because I needed to see the output for all 3 features to make predictions successfully on each. In addition, this helps the accuracy of the model for them to understand the relationships of these features together and how they affect one another. I used a learning rate of .01 and the methods of Mean Squared Error (MSE) and the Adam optimizer method. I trained the model with 100 epochs and then tested for validation and training loss. I then saved the best model with the best validation loss by using torch's save function. I did this so when I do create predictions I would have the potential to have the most accurate predictions possible. Then I graphed the training and validation loss over time in the training stages showcased in Figure 1 and Figure 2.

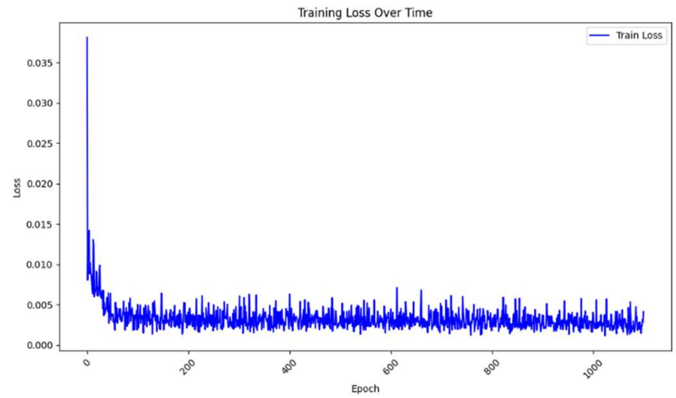

Figure 1. Training loss over time whilst training the LSTM model.

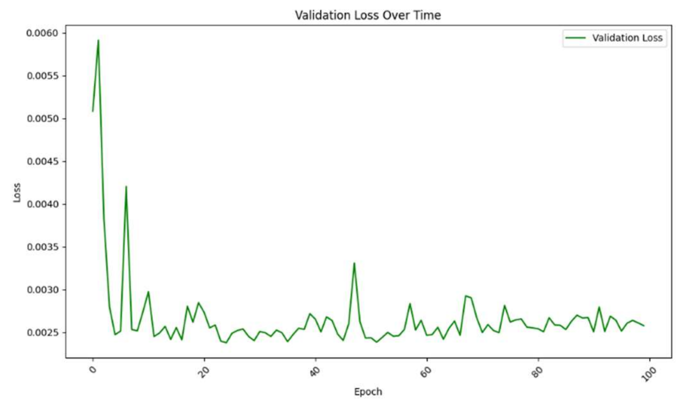

Figure 2. Validation loss over time whilst training the LSTM model.

As shown in Figure 1, it is apparent that in the beginning of training there was the most loss and over time is begins to oscillate from .005 to .0001. This can be concluded that from the learning rate in the model it began to have a steady pace at maintaining a certain level of loss. Moreover, Figure 2. showcases the validation loss is quite higher in the beginning of the training than all else for the most part. However, there is one of a little higher peak than most found at roughly close to the 50th epoch. In general, these first two figures show great promise to a well trained model with little loss.

D. Creating the Predictions

After training and testing the model, the next step is to do the predictions. I scaled the data for Moving Averages, Open Price, and Volume then put these features into the created sequences based on the window size. Then I convert the sequences to Pytorch tensors to input them to the model. After the model predicts these scaled values, I have to reverse the scale to see the predictions in their own original units. Lastly, I plot them to see how they compare to the actual data from the past five years.

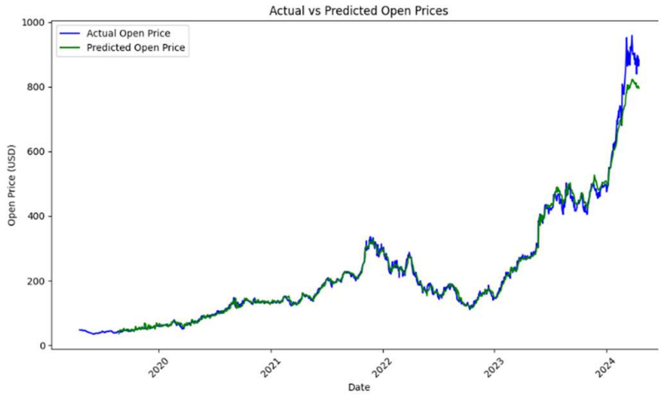

Figure 3. The Actual Open Price versus The LSTM Model's Predicted Open Price for the Stock Nvidia from April 2019 to April 2024.

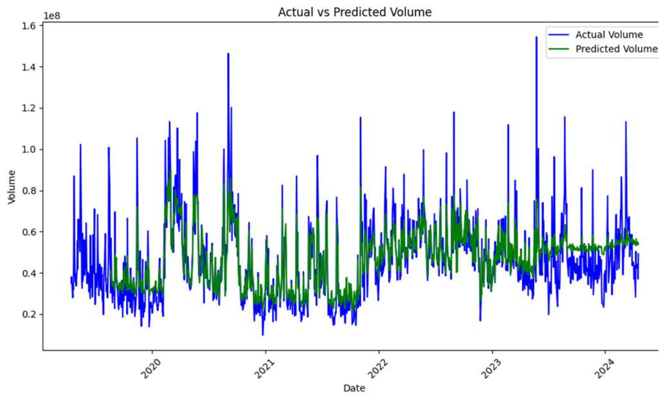

Figure 4. The Actual Volume versus Predicted Volume of the Nvidia Stock from April 2019 to April 2024.

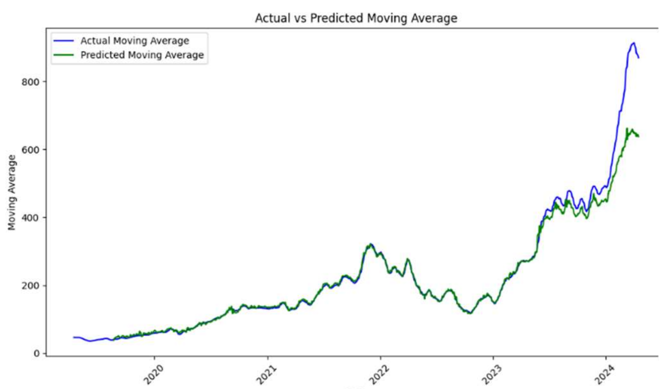

Figure 5. The Actual Moving Average (10 days) versus Predicted Moving Average (10 days) of the Nvidia Stock April 2019 to April 2024.

These are the results of the first prediction algorithm to visualize how well the model aligns compared to the actual data from the past five years.

Lastly, I wanted to see the predictions from 90 days into the future from April 15th, 2024 which roughly is estimating to July 15th, 2024. The methods of doing this is the same as the previous algorithm, but I take the last 90 days of our collected data to make the next predicted sequence to add to our current prediction line I already have. The following figures are the results of the actual 90 day prediction algorithm beginning its prediction to be April 16th, 2024.

Figure 6. The Actual Open Price and Predicted Open Price for the 90 days after April 15th, 2024.

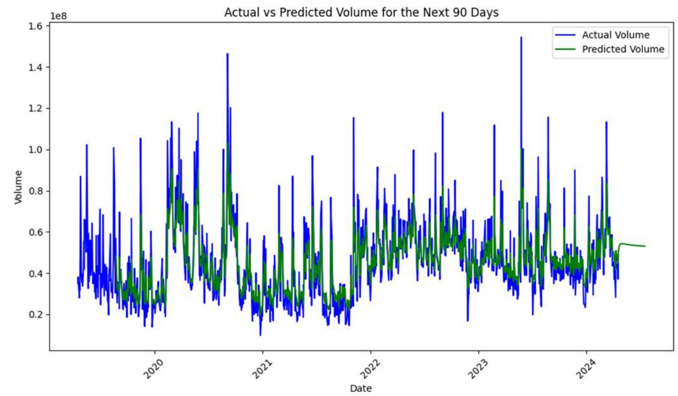

Figure 7. The Actual Volume and Predicted Volume for 90 days after April 15th, 2024.

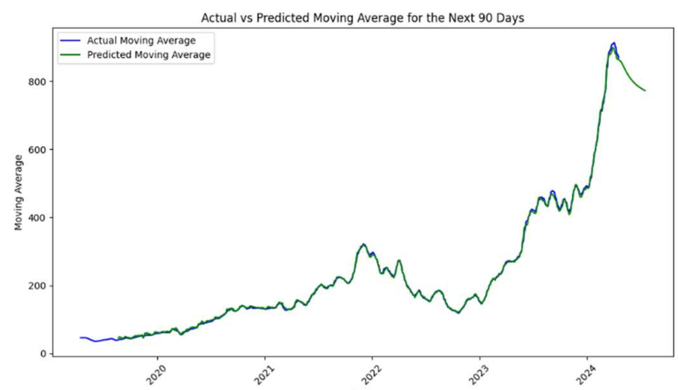

Figure 8. The Actual Moving Average (10) and the Predicted Moving Average (10) for 90 days after April 15th, 2024.

E. *Overall Coding Packages Used*

The packages I used to create the code for my model and predictions consisted of the following in the language Python: pandas, torch, matplotlib, and sklearn. In addition, I used many subclass packages of these modules.

IV. Results

Firstly, Figures 3, 4, and 5 visually showcase the difference between the actual and prediction lines from the past 5 years. The prediction line is generally very accurate until middle 2023. The prediction line appears the be on the actual line until the middle point of 2023 for figure 5 but more distinctively on the line until closer to the end of the data for figure 3. Figure 4 in comparison of accuracy to the others is the least accurate, but this is to be expected as the volume of people partaking in a stock is to be expected to be easy to predict by any means. Moreover, I would argue say the model displays very accurate tendencies for these figures 3, 4, and 5 in comparison to the actual data line.

For Figures 6, 7, and 8 I have our prediction for 90 days into the future. Given the connotation of Nvidia believed to increase because of its recent positive light in the news which lead more people to invest, my predictions could be taken lightly. Moreover, while the predictions are predicting more of a downward trend this is not extremely uncommon given the fact of the history of this stock having ups and downs as all stocks typically do in their own cycles. The model is likely predicting it to slightly downturn as the current price is very high and it has experienced downturns such as 2022 to 2023. Overall, I would say these results are fairly a success as it definitely could plausibly happen as the stock market is extremely hard to predict.

One important thing to note is that this model can be saved if it was extremely accurate on its run, it would just need to be specified to be saved. However, every time this code is ran the training is different; therefore, the results and predictions will be different. This could be considered to be a drawback on the model. This could be one thing to be improved to find a more stable hyperparameters potentially to give close to the same results each time it is ran.

V. Conclusion

In conclusion, I believe this model predicts as well as it can without implementing other outside factors that affects stock price. I believe it would be extremely more useful if, as stated in the motivation, factors of monetary and fiscal policy were included into the stock prediction. Moreover, the factor of implementing a feature representation of the current news surrounding the stock would be also ideal. However, there may be limitations on that exactly to train a model on attaining Likert Scale data of how well they are doing in the news respective to the average investor. In general, I believe the neural network could be improved by adding more features to give more insight into the complex factors affecting the stock market.

The biggest difficulty of this project was the trial and error of creating a accurate model. At first, I put so much time into making a one input model based on open price only and it did not predict accurately as it did not account for many things. I even spent hours making a Grid Search algorithm to find the best hyperparameters to fix it and it still was not successful as it was not accounting for enough other factors. Another difficulty of this project was I had accidentally retrained the model before making a copy of the results in Google Collab, so I had lost the exact model that produced these results. However, these results happen to be the best results the model produces. This project taught me so much about the small differences of results just even retraining the model could give. I believe this would be less affected though if there was more data than 5 years and perhaps a smaller window size to predict into the future. Furthermore, the results would likely be less affected if a certain combination of steady hyper parameters were achieved.

The biggest successes I have gained from throughout this project would include the initiative of researching and learning into how to begin an LSTM stock prediction model. Moreover, I felt I was extremely successful with the prediction line of the past 5 years. The model could generally improve to predict the next 90 days better, if once again there are more features added.

Future work of this project would be taking the steps to make this model more complex. I would attempt to try to make the better model with even more statistical features along with economic indicator features. More implications of making future work for this project would be also accounting for other popular stocks at the same time and perhaps the effects other stock prices put onto this one etc. This would prove to be a challenging task but it definitely could be possible. Lastly, more improvements could be suggested by asking for input from Economic professors or people who work every day in the stock market with immense knowledge of the mechanics of how things work.

Acknowledgment

I would like to give thanks and acknowledgement to my professor, Dr. Yu Liang.

References

[1] "NVIDIA Corporation (NVDA) Stock Price, News, Quote & History - Yahoo Finance." *Finance.yahoo.com*, 2024, finance.yahoo.com/quote/NVDA?.tsrc=fin-srch.

[2] "Stock Market Analysis + Prediction Using LSTM." *Kaggle.com*, www.kaggle.com/code/faressayah/stock-market-analysis-prediction-using-lstm/notebook.